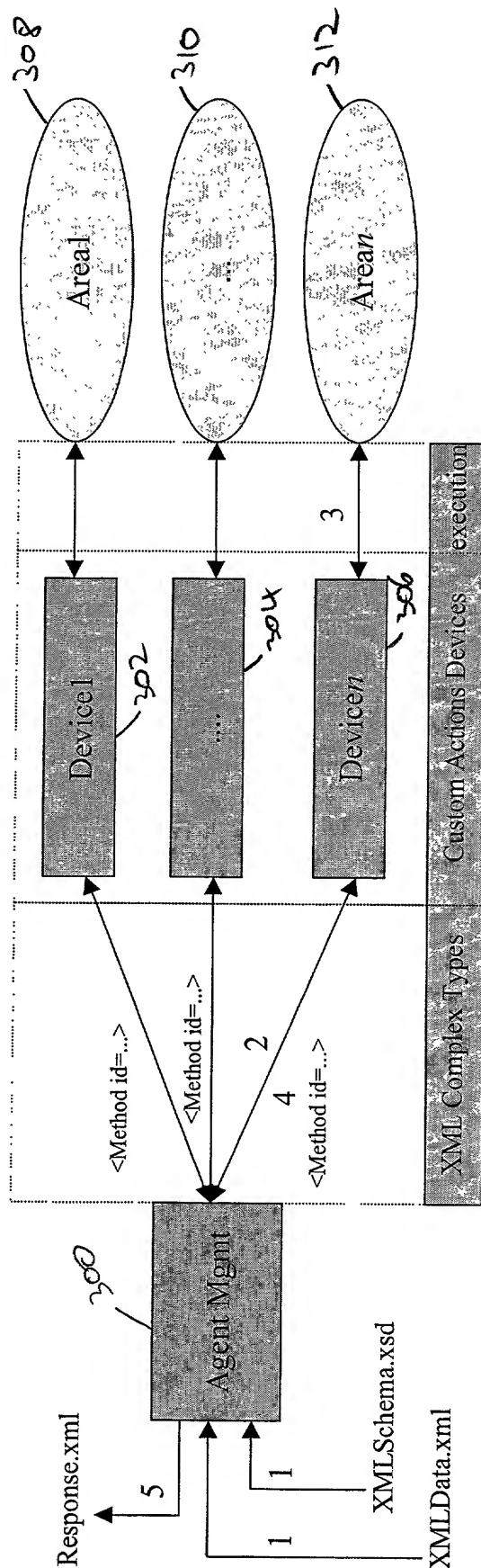


Architecture overview



- 1) Request XML file is passed to Agent Mgmt containing complex types
 Agent Mgmt splits the data into parts and hands the <Method> over to the corresponding Custom Actions Devices
- 2) The complex type <Method> describes the method to execute and its parameters
- 3) Custom Actions Devices execute the functions in a certain area they are responsible for and pack the result into XML complex types
- 4) Complex types containing result data are returned to Agent Mgmt
- 5) XML stream has been packed and is returned

Fig. 1

```

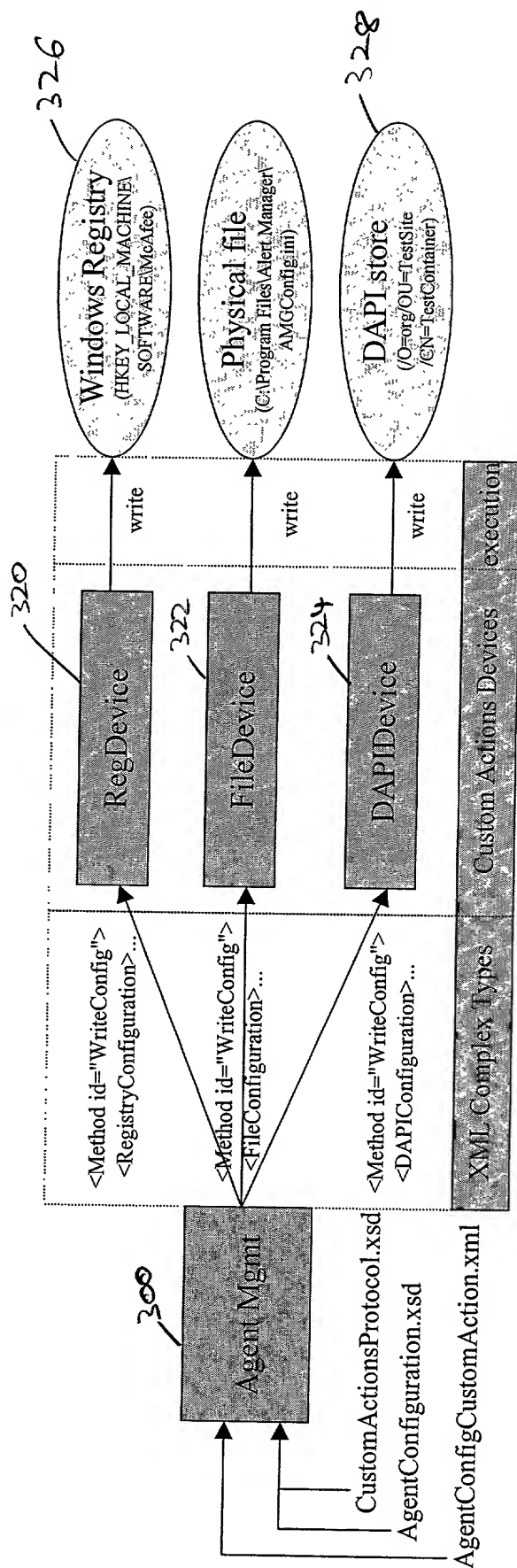
graph TD
    C1[CustomActionsProtocolReq.xml] -- 1 --> AM[Agent Mgmt]
    C2[CustomActionsProtocol.xsd] -- 1 --> AM
    AM -- 5 --> C3[CustomActionsProtocolResp.xml]
    AM -- 1 --> XCT[XML Complex Types]
    XCT -- 2 --> APD[API Device]
    XCT -- 4 --> ID[Install Device]
    XCT -- 3 --> ED[Event Device]
    APD -- 314 --> W32[Win32 API function]
    ID -- 316 --> SIP[Software install packages]
    ED -- 318 --> EAC[Eventing/Alerting components]
    subgraph CAD [Custom Actions Devices]
        W32
        SIP
        EAC
    end

```

- 1) Request XML file is passed to Agent Mgmt containing complex types
- 2) The complex type <Method> describes the method to execute and its parameters
Each <Method> is passed to the corresponding Custom Actions Device
- 3) Custom Actions Devices execute the functions and pack the result into XML complex types
- 4) Complex types containing result data are returned to Agent Mgmt
- 5) XML stream has been packed and is returned

2
Li⁺

Deploying Configuration Data



- 1) Agent Mgmt receives AgentConfigCustomAction.xml
- 2) .xml file is validated against .xsd file(s) to make sure valid data is written
- 3) XML Complex types are sent to Custom Actions Devices, the parameters containing the configuration data
- 4) Custom Actions Devices update the config store they are responsible for
- 5) Optionally a return value can be returned as a Response.xml file

Fig. 3

Retrieving Configuration Data

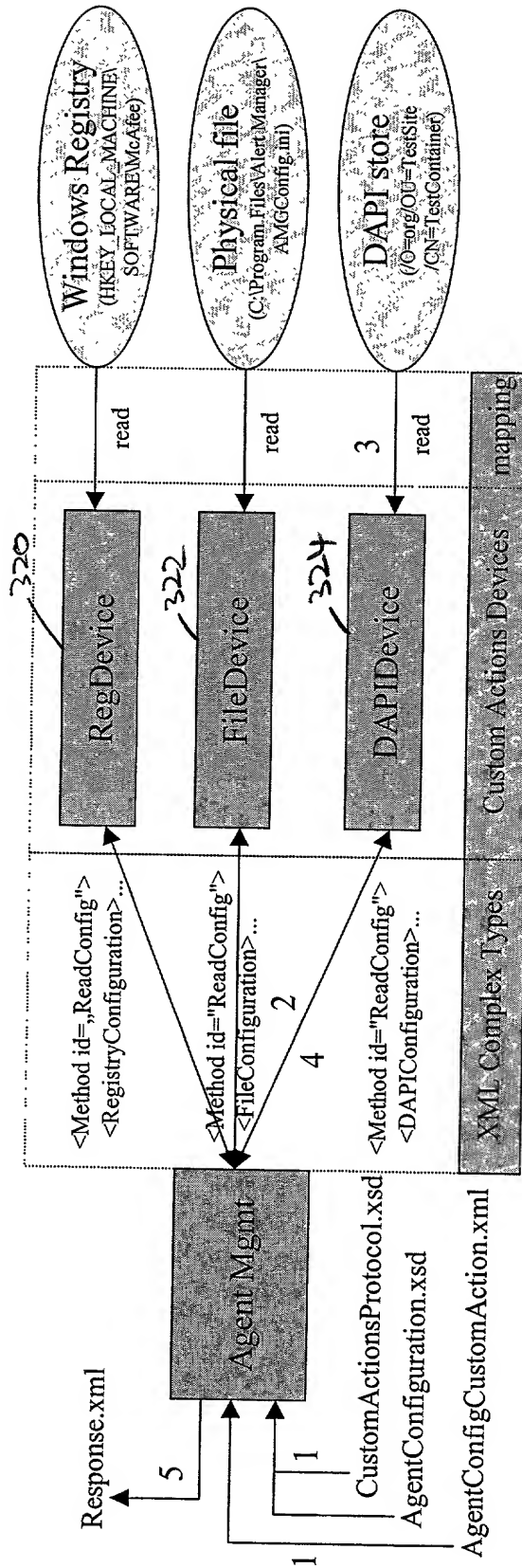


Fig. 4

Initiator

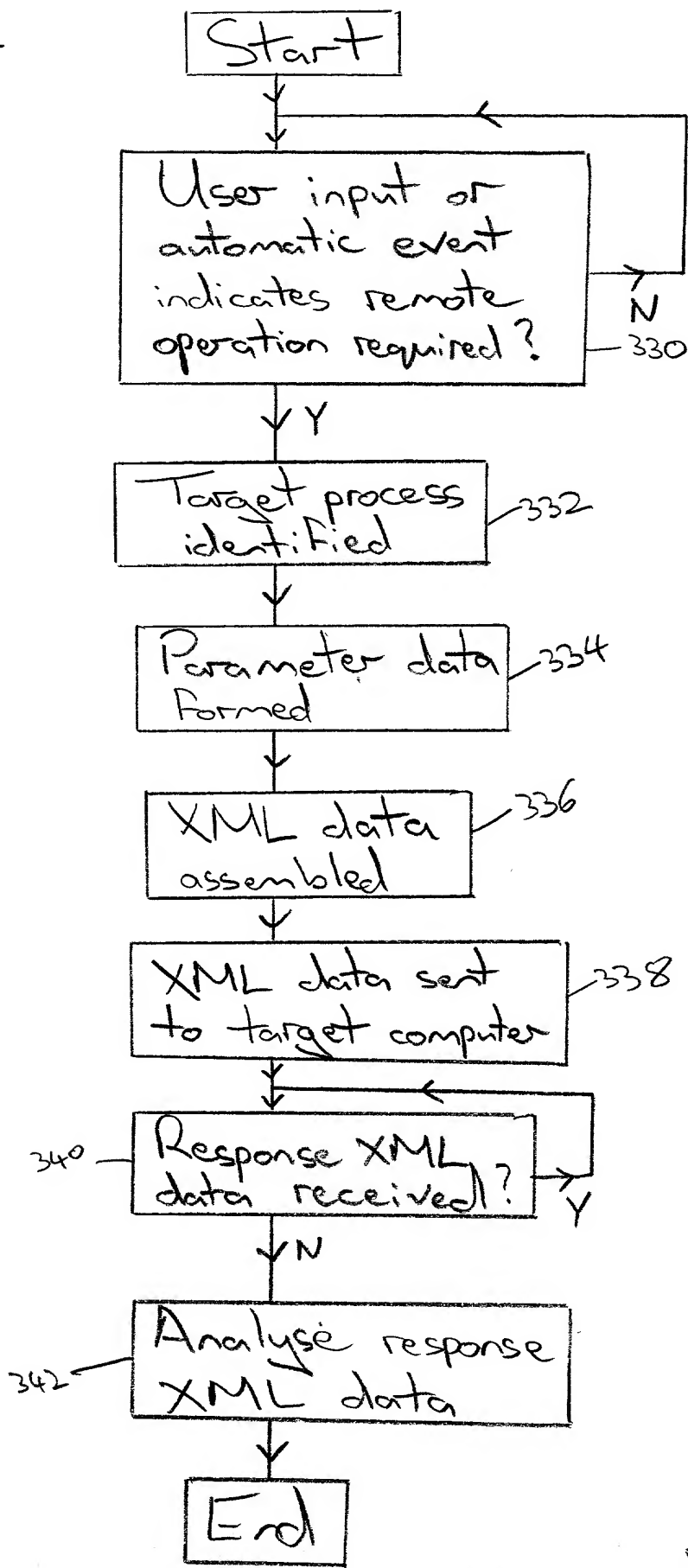


Fig. 5

Agent

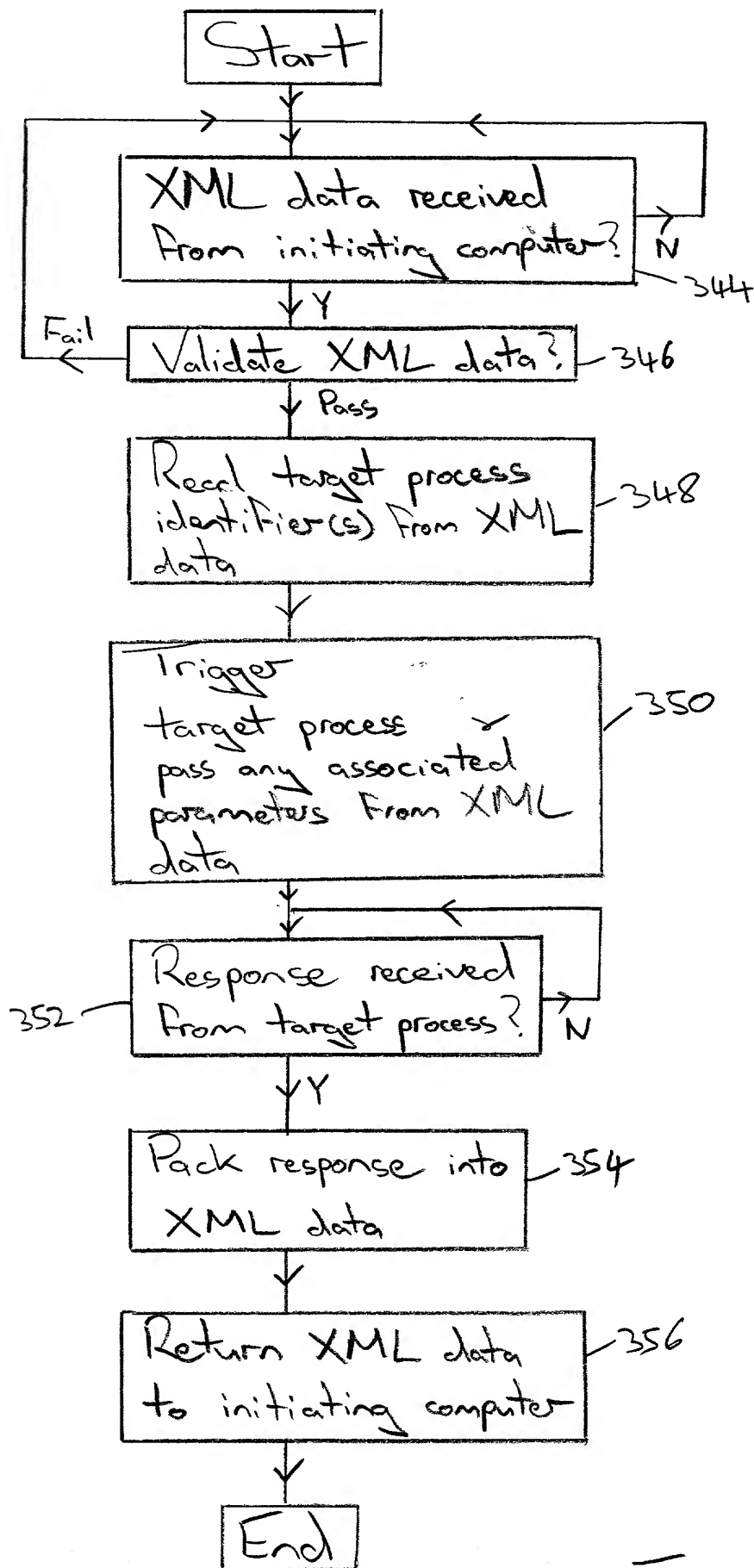


Fig. 6

Target

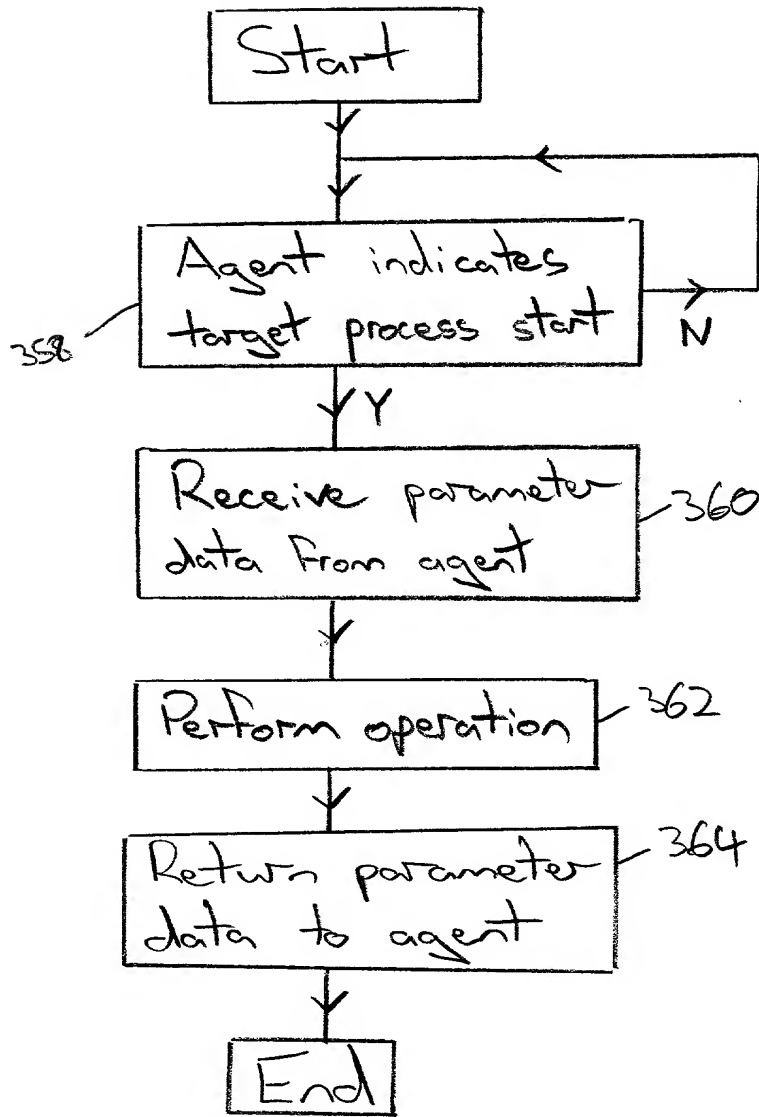
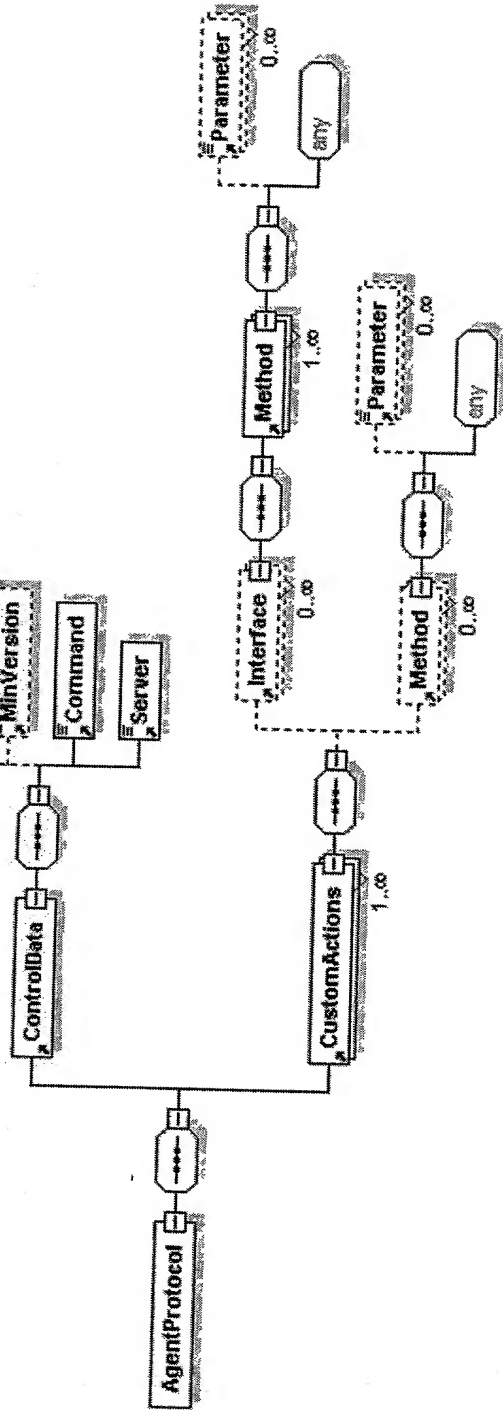


Fig. 7

20220601 12:26:00



CustomActionsProtocol.xsd

Fig. 9

```

<?xml version="1.0" ?>
- <AgentProtocol xmlns="http://www.nai.com"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.nai.com CustomActionsProtocol.xsd">
- <ControlData>
  <Version>0x01000001</Version>
  <MinVersion>0x01000001</MinVersion>
  <Command>RequestCustomAction</Command>
  <Server>ned1wnts2ke</Server>
</ControlData>
- <CustomActions
  id="<AGENT_INSTALLED_DIR>\\CustomActionsLibrary\\CustAct1.dll">
- <Method id="GetRegStringValue">
  <Parameter id="Key" type="xs:string"
    inout="in"><AGENT_INSTALLED_REGKEY></Parameter>
  <Parameter id="Valuename" type="xs:string"
    inout="in">AgentVersion</Parameter>
  <Parameter id="Result" type="xs:string" inout="out" />
</Method>
</CustomActions>
- <CustomActions id="{06E0062A-5069-4793-ACED-F80BE1BBC4AF}">
- <Interface id="{C9E1CC03-8007-412A-8F5D-532C57DF4482}">
- <Method id="ExecuteSilentInstallation">
  <Parameter id="ProductName" type="xs:string"
    inout="in">TestInstallProduct</Parameter>
  <Parameter id="ProductVersion" type="xs:decimal"
    inout="in">0x01000001</Parameter>
  <Parameter id="Location" type="xs:string"
    inout="in">c:\InstallImages</Parameter>
  <Parameter id="Result" type="xs:string" inout="out" />
</Method>
</Interface>
- <Interface id="{C9E1CC03-8007-412A-8F5D-532C57DF4482}">
- <Method id="GetSystemDirectory">
  <Parameter id="Directory" type="xs:string" inout="out" />
  <Parameter id="Result" type="xs:decimal" inout="out" />
</Method>
</Interface>
</CustomActions>
- <CustomActions id="{06E0062B-5069-4793-ACED-F80BE1BBC4AF}">
- <Interface id="{A000CC03-8007-412A-8F5D-532C57DF4482}">
- <Method id="TriggerEvent">
  <Parameter id="EventID" type="xs:decimal"
    inout="in">1000</Parameter>
  <Parameter id="EventDescription" type="xs:decimal"
    inout="in">The event %EventID% has been triggered by %
    USERNAME% on computer %COMPUTERNAME%. The %
    FILENAME% file is infected with %VIRUSNAME%. This has
    been detected by engineversion %ENGINEVERSION%
    datversion %DATVERSION%.</Parameter>
  <Parameter id="COMPUTERNAME" type="xs:string"
    inout="in">sourcecomputer</Parameter>
  <Parameter id="USERNAME" type="xs:string"
    inout="in">sourceuser</Parameter>
  <Parameter id="FILENAME" type="xs:string"
    inout="in">kernel32.dll</Parameter>
  <Parameter id="VIRUSNAME" type="xs:string"

```

Fig. 10A

```
inout="in">Nimbda</Parameter>
  <Parameter id="ENGINEVERSION" type="xs:decimal"
    inout="in">0x04005001</Parameter>
  <Parameter id="DATVERSION" type="xs:decimal"
    inout="in">0x07003009</Parameter>
  <Parameter id="Result" type="xs:string" inout="out" />
</Method>
</Interface>
</CustomActions>
</AgentProtocol>
```

103544-000

Fig. 103

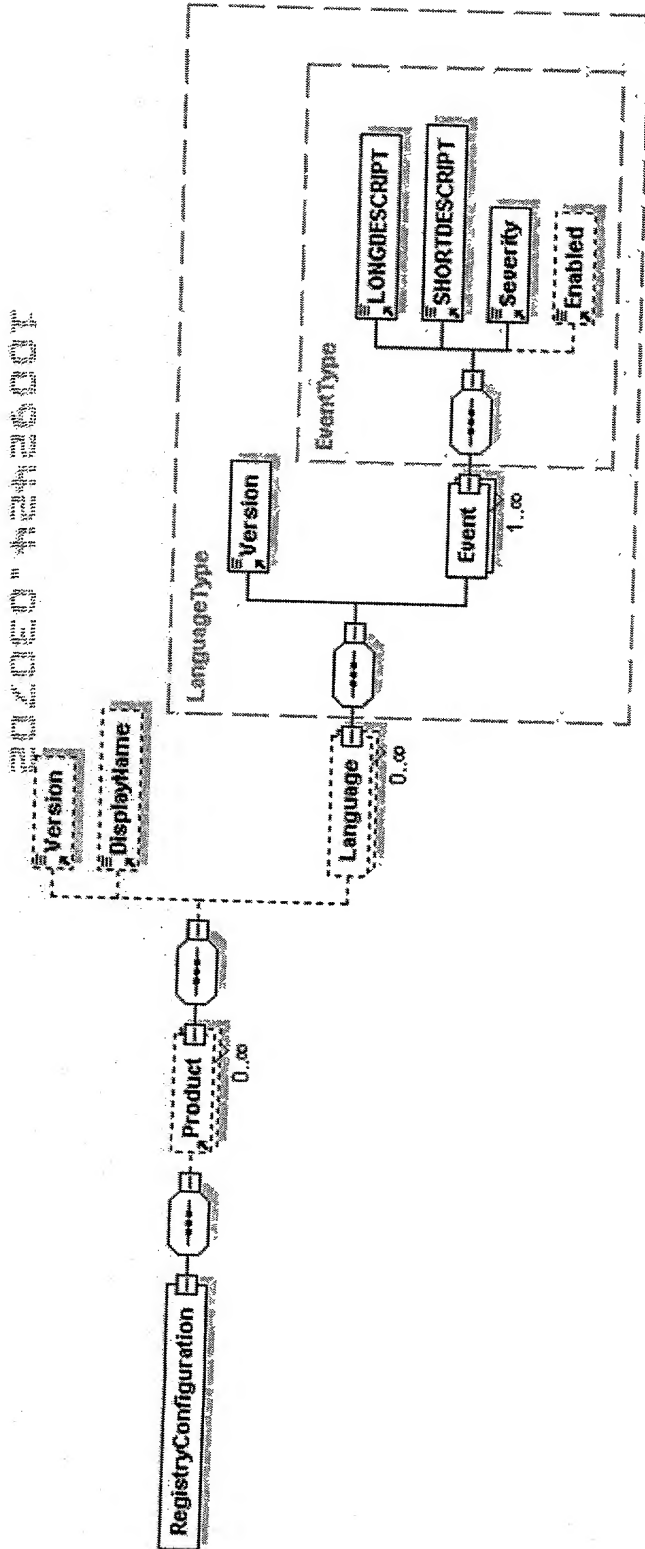
```

<?xml version="1.0" ?>
- <AgentProtocol xmlns="http://www.nai.com"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.nai.com CustomActionsProtocol.xsd">
- <ControlData>
  <Version>0x01000001</Version>
  <MinVersion>0x01000001</MinVersion>
  <Command>RespondToCustomAction</Command>
  <Server>ned1wnts2ke</Server>
</ControlData>
- <CustomActions
  id="<AGENT_INSTALLED_DIR>\\CustomActionsLibrary\\CustAct1.dll">
- <Method id="GetRegStringValue">
  <Parameter id="Result" type="xs:string"
    inout="out">5.0.1.10</Parameter>
</Method>
</CustomActions>
- <CustomActions id="{06E0062A-5069-4793-ACED-F80BE1BBC4AF}">
- <Interface id="{C9E1CC03-8007-412A-8F5D-532C57DF4482}">
  - <Method id="ExecuteSilentInstallation">
    <Parameter id="Result" type="xs:string" inout="out">Error: Invalid
      Image path specified.</Parameter>
  </Method>
</Interface>
- <Interface id="{C9E1CC03-8007-412A-8F5D-532C57DF4482}">
  - <Method id="GetSystemDirectory">
    <Parameter id="Directory" type="xs:string"
      inout="out">C:\Winnt\System32</Parameter>
    <Parameter id="Result" type="xs:decimal"
      inout="out">0</Parameter>
  </Method>
</Interface>
</CustomActions>
- <CustomActions id="{06E0062B-5069-4793-ACED-F80BE1BBC4AF}">
- <Interface id="{A000CC03-8007-412A-8F5D-532C57DF4482}">
  - <Method id="TriggerEvent">
    <Parameter id="Result" type="xs:string" inout="out">Event sent to
      testcomputer2</Parameter>
  </Method>
</Interface>
</CustomActions>
</AgentProtocol>

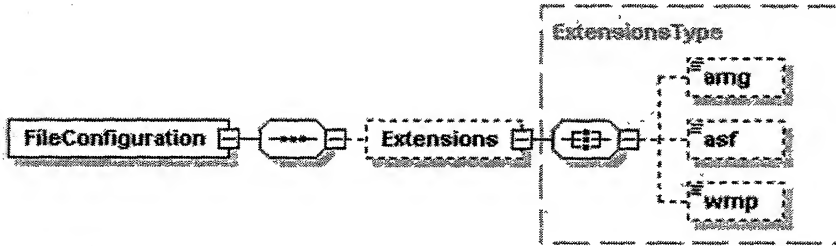
```

Fig. 11

Fig. 12



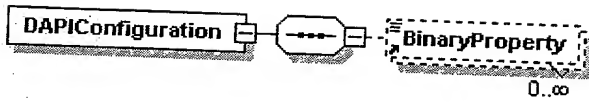
Agent Configuration.xsd - Registry Configuration



Agent Configuration.xsd - FileConfiguration

20/06/2014 14:24:00

Fig. 13



AgentConfiguration.xsd - DAPIConfiguration

2007-04-24 14:26:00

Fig. 14

AgentConfigCustomAction.xml

Inventor: NEDBAL, M.
SN unknown/Sheet 16 of 27
Atty. Dkt.: 550-320

```
<?xml version="1.0" ?>
- <AgentProtocol xmlns="http://www.nai.com"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.nai.com CustomActionsProtocol.xsd
    http://www.nai.com AgentConfiguration.xsd">
- <ControlData>
  <Version>0x01000001</Version>
  <MinVersion>0x01000001</MinVersion>
  <Command>RequestCustomAction</Command>
  <Server>ned1wnts2ke</Server>
</ControlData>
- <CustomActions id="RegistryMapping.dll">
- <Method id="WriteConfig">
  - <RegistryConfiguration
    id="HKEY_LOCAL_MACHINE\SOFTWARE\McAfee">
  - <Product id="Alert Manager">
    <Version>0x04070000</Version>
    <DisplayName>Alert Manager 4.7</DisplayName>
  - <Language id="0407">
    <Version>0x01000002</Version>
    - <Event id="1">
      <LONGDESCRIPT>Das ist eine Test-Nachricht von Alert
        Manager.</LONGDESCRIPT>
      <SHORTDESCRIPT>Testing</SHORTDESCRIPT>
      <Severity>5</Severity>
      <Enabled>1</Enabled>
    </Event>
  </Language>
  - <Language id="0409">
    <Version>0x01000002</Version>
    - <Event id="1">
      <LONGDESCRIPT>This is an alert manager test
        messge.</LONGDESCRIPT>
      <SHORTDESCRIPT>Testing</SHORTDESCRIPT>
      <Severity>0</Severity>
      <Enabled>1</Enabled>
    </Event>
    - <Event id="2">
      <LONGDESCRIPT>Text of event 2.</LONGDESCRIPT>
      <SHORTDESCRIPT>Testing</SHORTDESCRIPT>
      <Severity>1</Severity>
    </Event>
  </Language>
  </Product>
</RegistryConfiguration>
</Method>
- <Method id="ReadConfig">
  <RegistryConfiguration
    id="HKEY_LOCAL_MACHINE\SOFTWARE\McAfee\*" />
</Method>
</CustomActions>
- <CustomActions id="INIFileMapping.dll">
- <Method id="WriteConfig">
  - <FileConfiguration id="C:\Program Files\Alert
    Manager\AMGConfig.ini">
  - <Extensions>
```

Fig. 15A

[illegible]

Fig. 15B

Source

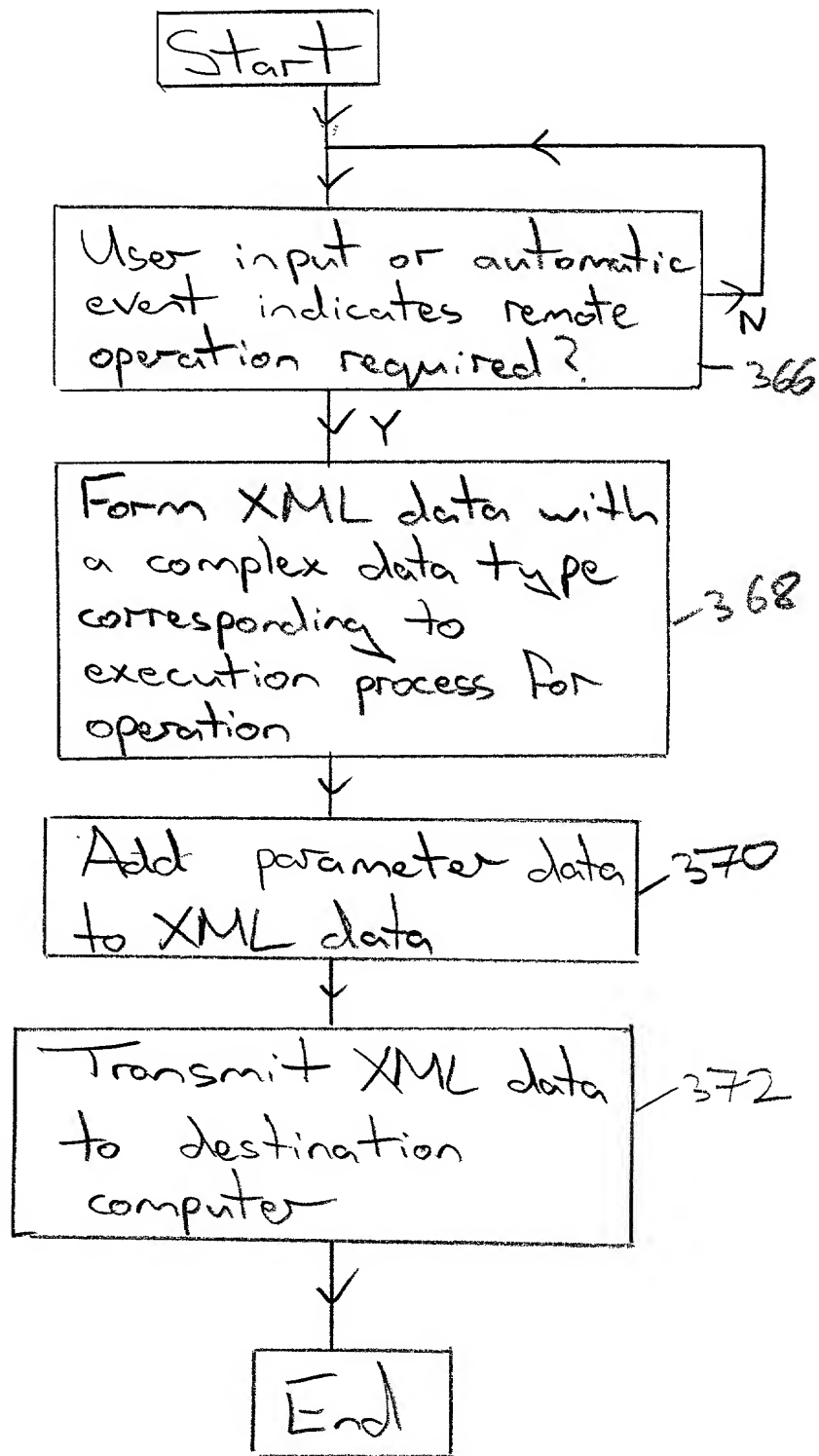


Fig. 16

Destination

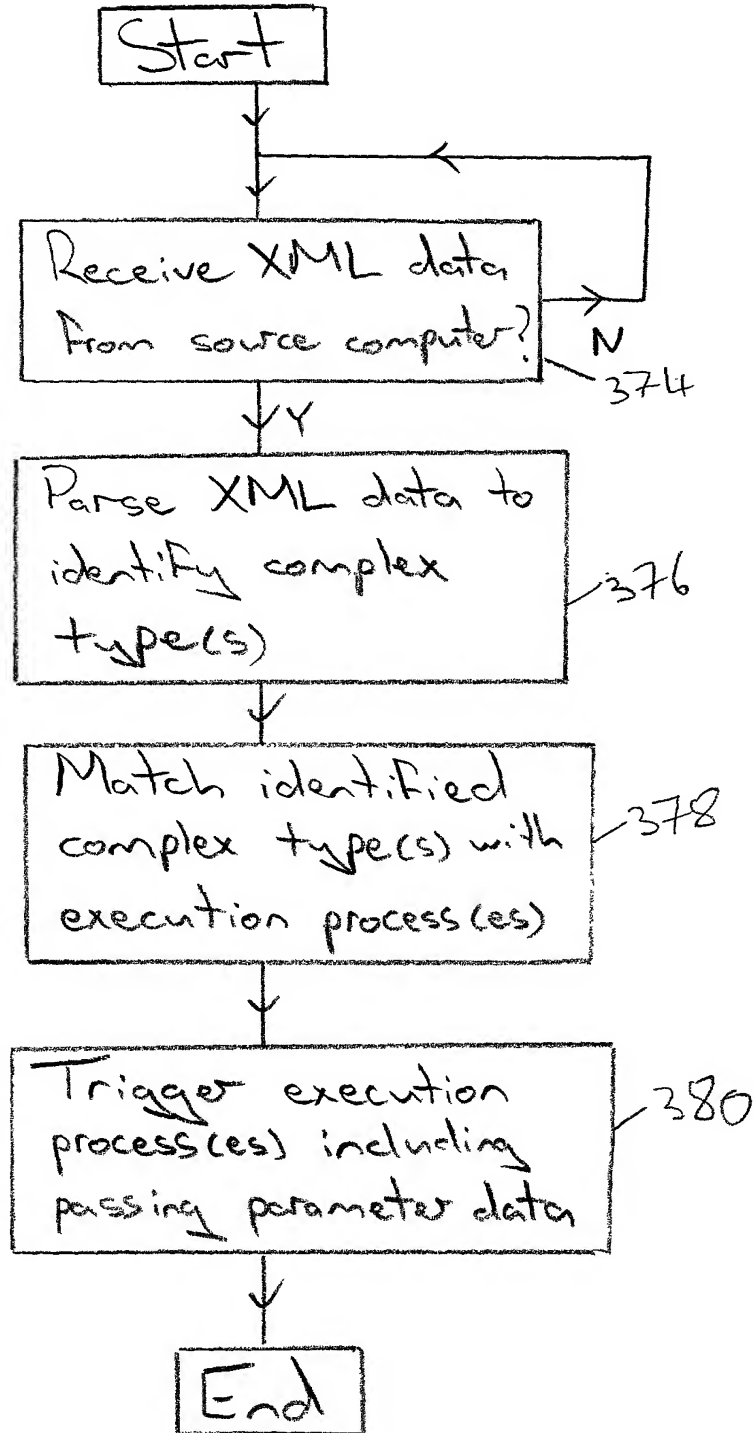


Fig. 17

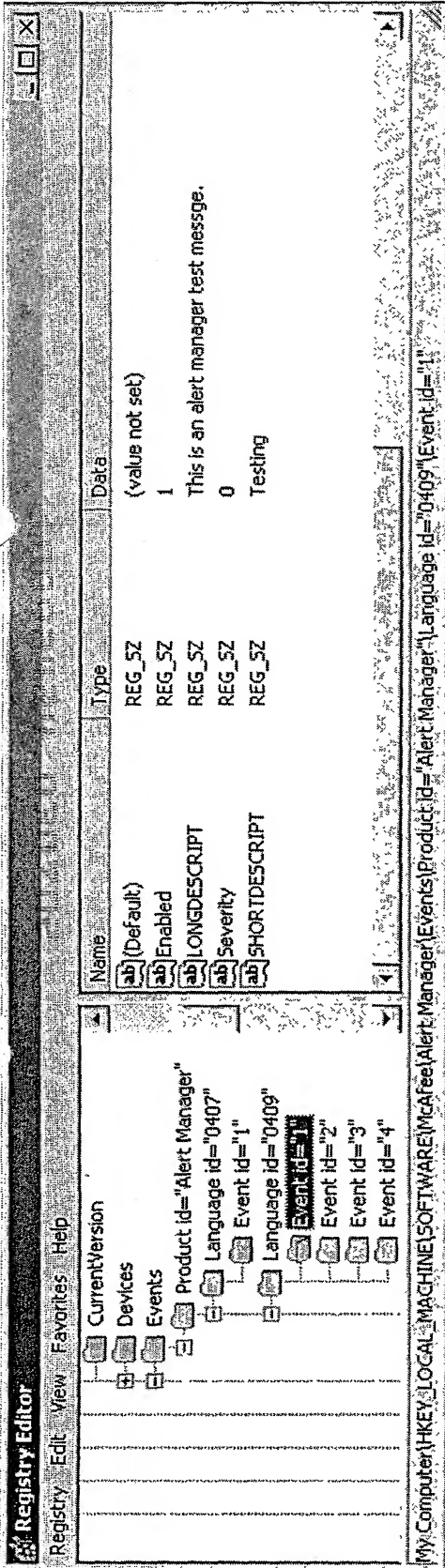


Fig. 18

Registry Data

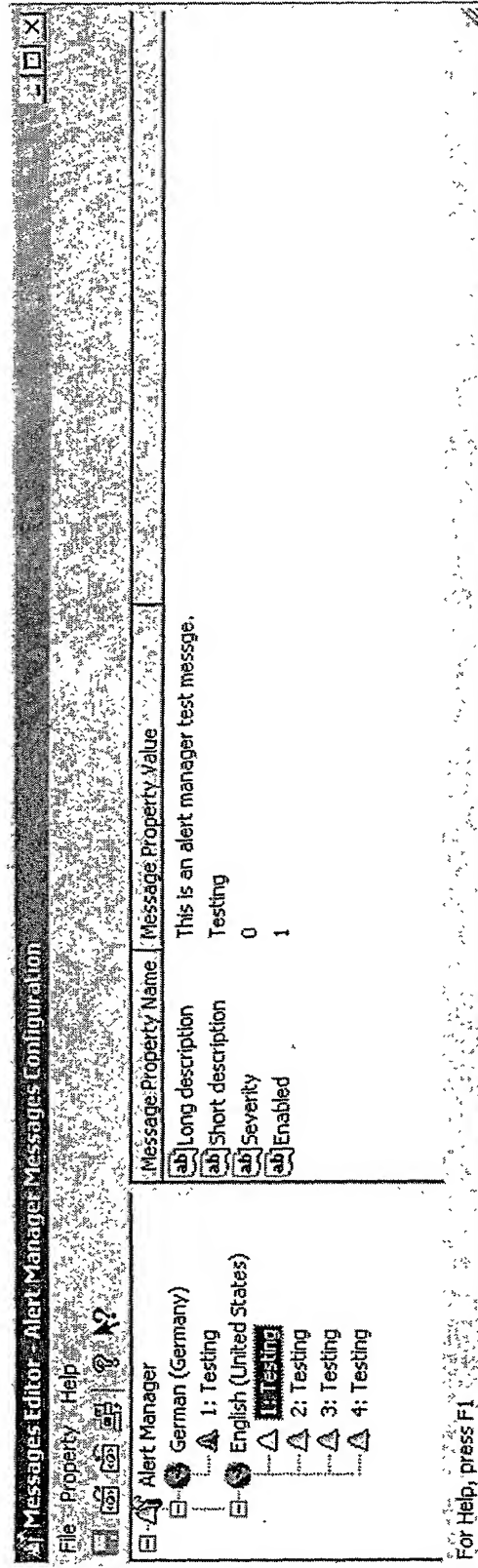


Fig. 19

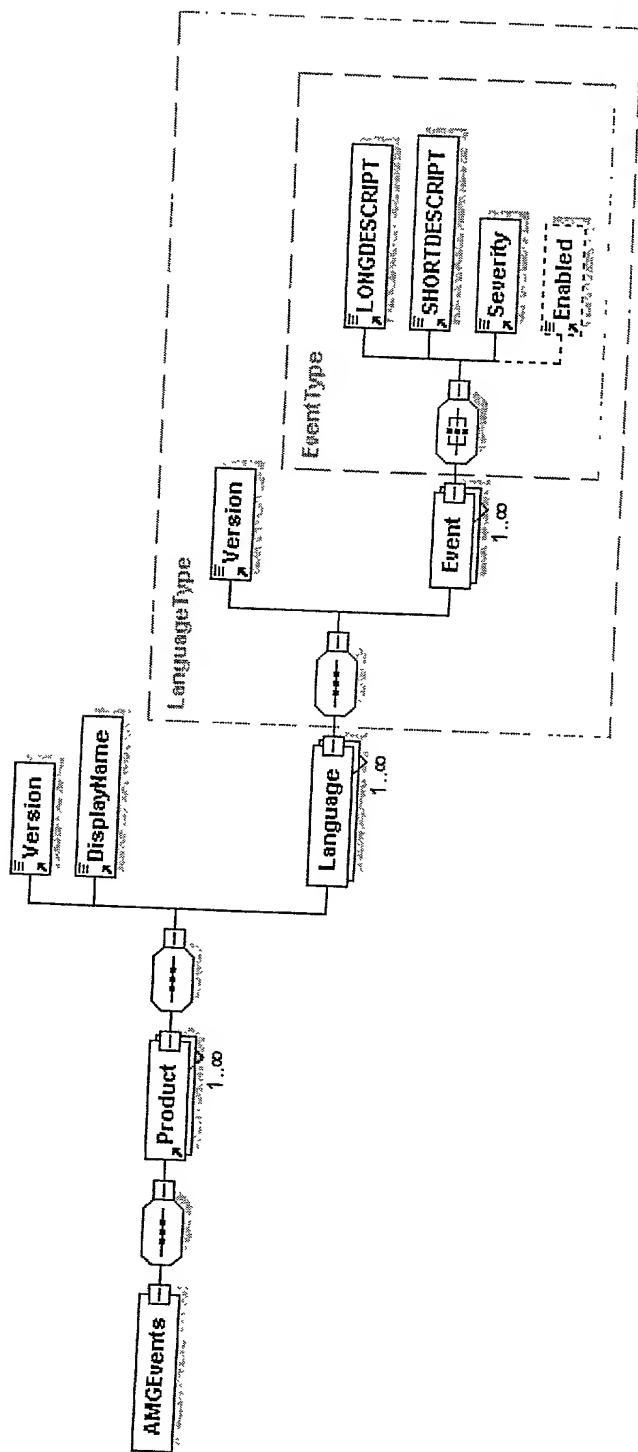
DOM Data View

<?xml version="1.0" ?>
- <AMGEvents xmlns="http://www.nai.com"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.nai.com AMGEvents.xsd">
 - <Product id="Alert Manager">
 <Version>0x04070000</Version>
 <DisplayName>Alert Manager 4.7</DisplayName>
 - <Language id="0407">
 <Version>0x01000002</Version>
 - <Event id="1">
 <LONGDESCRIPT>Das ist eine Test-Nachricht von Alert
 Manager.</LONGDESCRIPT>
 <SHORTDESCRIPT>Testing</SHORTDESCRIPT>
 <Severity>5</Severity>
 <Enabled>1</Enabled>
 </Event>
 </Language>
 - <Language id="0409">
 <Version>0x01000002</Version>
 - <Event id="1">
 <LONGDESCRIPT>This is an alert manager test
 messge.</LONGDESCRIPT>
 <SHORTDESCRIPT>Testing</SHORTDESCRIPT>
 <Severity>0</Severity>
 <Enabled>1</Enabled>
 </Event>
 - <Event id="2">
 <LONGDESCRIPT>Text of event 2.</LONGDESCRIPT>
 <SHORTDESCRIPT>Testing</SHORTDESCRIPT>
 <Severity>1</Severity>
 </Event>
 - <Event id="3">
 <LONGDESCRIPT>Text of event 3.</LONGDESCRIPT>
 <SHORTDESCRIPT>Testing</SHORTDESCRIPT>
 <Severity>1</Severity>
 </Event>
 - <Event id="4">
 <LONGDESCRIPT>Text of event 4.</LONGDESCRIPT>
 <SHORTDESCRIPT>Testing</SHORTDESCRIPT>
 <Severity>1</Severity>
 </Event>
 </Language>
</Product>
</AMGEvents>

XML Data

Fig. 20

20/03/2014 14:24:53



Generated with XMLSpy Schema Editor www.xmlspy.com

XSD Data

Fig. 21

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- edited with XML Spy v4.0.1 J (http://www.xmlspy.com) by Napalm.
(Napalm) -->
- <xs:schema targetNamespace="http://www.nai.com"
  xmlns="http://www.nai.com"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
  <xs:element name="DisplayName" type="xs:string" />
  <xs:element name="Enabled" type="xs:boolean" />
  - <xs:complexType name="EventType">
    - <xs:all>
      <xs:element ref="LONGDESCRIPT" />
      <xs:element ref="SHORTDESCRIPT" />
      <xs:element ref="Severity" />
      <xs:element ref="Enabled" minOccurs="0" />
    </xs:all>
    <xs:attribute name="id" type="xs:string" use="required" />
  </xs:complexType>
  - <xs:complexType name="LanguageType">
    - <xs:sequence>
      <xs:element ref="Version" />
      <xs:element name="Event" type="EventType"
        maxOccurs="unbounded" />
    </xs:sequence>
    <xs:attribute name="id" type="xs:string" use="required" />
  </xs:complexType>
  - <xs:element name="Product">
    - <xs:complexType>
      - <xs:sequence>
        <xs:element ref="Version" />
        <xs:element ref="DisplayName" />
        <xs:element name="Language" type="LanguageType"
          maxOccurs="unbounded" />
      </xs:sequence>
      <xs:attribute name="id" type="xs:string" use="required" />
    </xs:complexType>
  </xs:element>
  - <xs:element name="AMGEvents">
    - <xs:complexType>
      - <xs:sequence>
        <xs:element ref="Product" maxOccurs="unbounded" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="LONGDESCRIPT" type="xs:string" />
  <xs:element name="SHORTDESCRIPT" type="xs:string" />
  <xs:element name="Severity" type="xs:string" />
  <xs:element name="Version" type="xs:string" />
</xs:schema>
```

XSD Data

Fig. 22

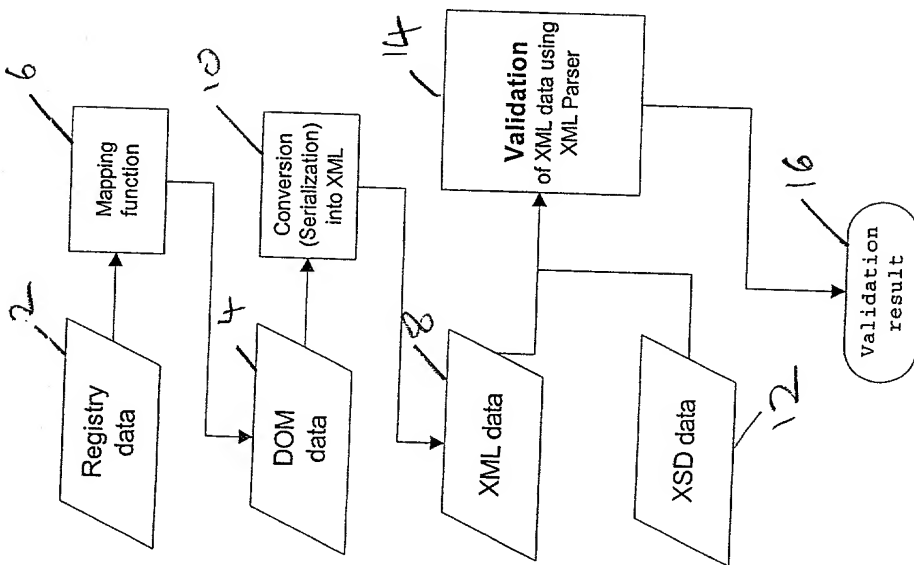


Fig. 23

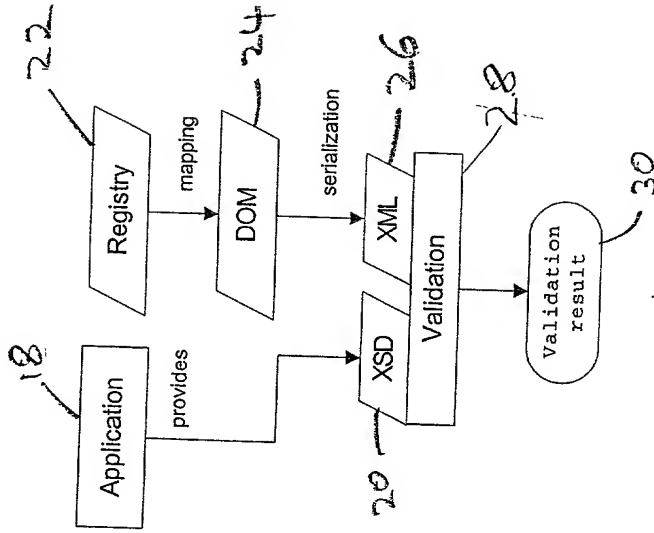


Fig. 24

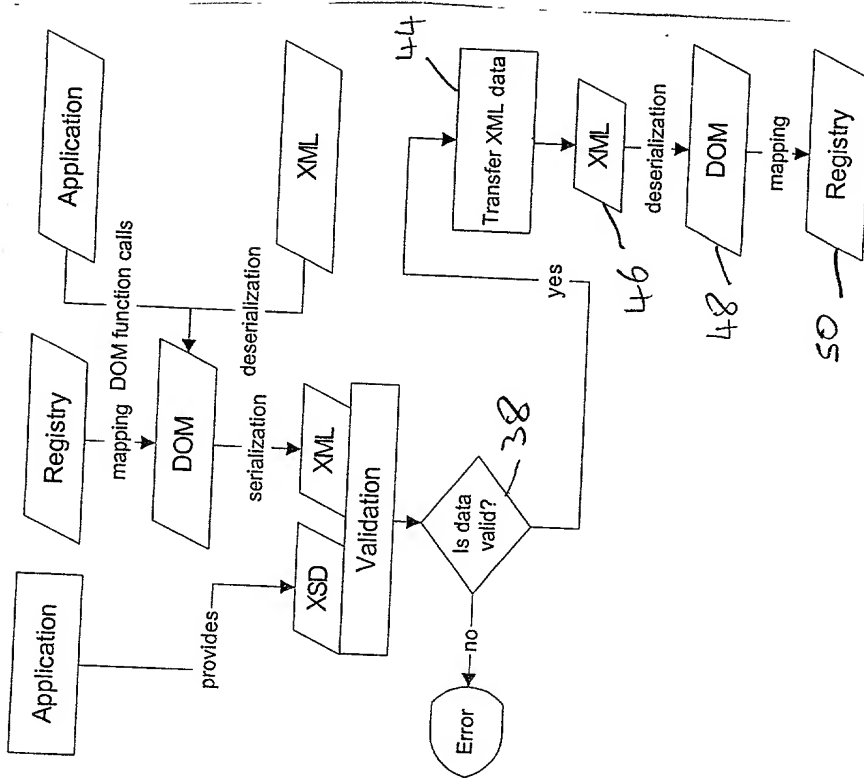


Fig. 26

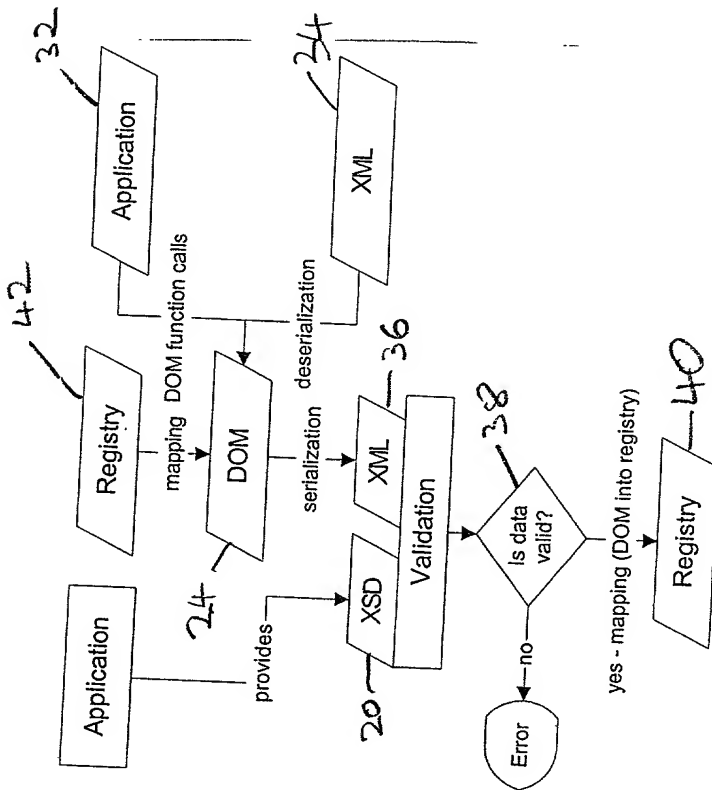


Fig. 25

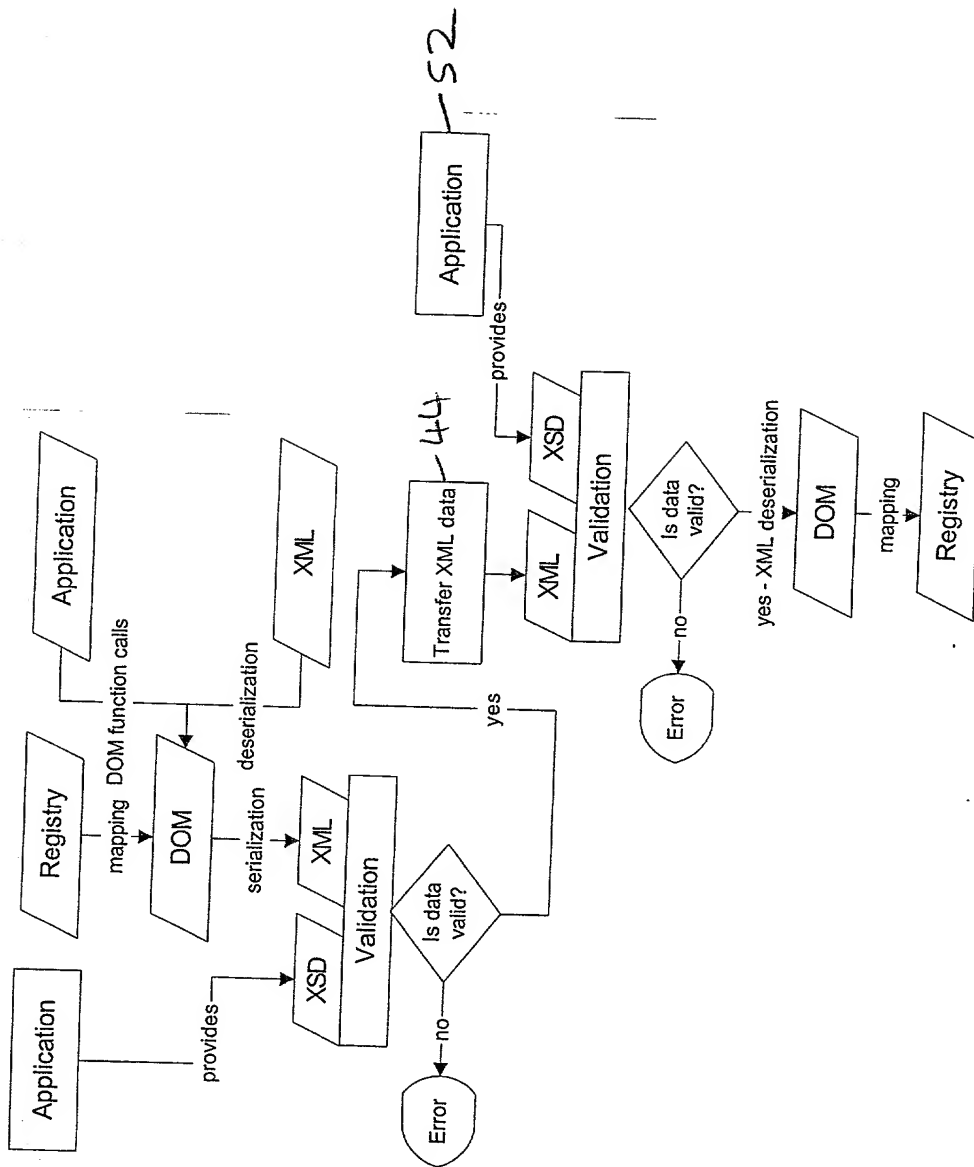


Fig. 27

